AF/2177

COMBINED TRANSMITTAL OF APPEAL BRIEF TO THE BOARD OF PATENT APPEALS AND INTERFERENCES & PETITION FOR EXTENSION OF TIME UNDER 37 C.F.R. 1.136(a) (Small Entity)	Docket No. WSS-10402/29

In Re Application Of: McMillan et al

Serial No.	Filing Date	Examiner	Group Art Unit
09/189,559	Nov. 11, 1998	Channavajjala	2177

Invention: METHOD AND SYSTEM FOR MANAGING SOFTWARE CONFLICTS AND COMPUTER-READABLE STORAGE MEDIUM HAVING A PROGRAM FOR EXECUTING THE METHOD

TO THE ASSISTANT COMMISSIONER FOR PATENTS:

This is a combined Transmittal of Appeal Brief to the Board of Patent Appeals and Interferences and petition under the provisions of 37 CFR 1.136(a) to extend the period for filing an Appeal Brief.

Applicant(s) hereby request(s) an extension of time of (check desired time period):

☒ One month ☐ Two months ☐ Three months ☐ Four months ☐ Five months

from: Sept. 21, 2003 (Date) until: Oct. 21, 2003 (Date)

The fee for the Appeal Brief and Extension of Time has been calculated as shown below:

Fee for Appeal Brief:	$165.00
Fee for Extension of Time:	$55.00
TOTAL FEE FOR APPEAL BRIEF AND EXTENSION OF TIME:	**$220.00**

The fee for the Appeal Brief and extension of time is to be paid as follows:

☒ A check in the amount of **$220.00** for the Appeal Brief and extension of time is enclosed.

☐ Please charge Deposit Account No. **07-1180** in the amount of
A duplicate copy of this sheet is enclosed.

☒ The Commissioner is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to Deposit Account No. **07-1180**
A duplicate copy of this sheet is enclosed.

- ☒ Any additional filing fees required under 37 C.F.R. 1.16.
- ☐ Any patent application processing fees under 37 CFR 1.17.

☒ If an additional extension of time is required, please consider this a petition therefor and charge any additional fees which may be required to Deposit Account No. **07-1180** A duplicate copy of this sheet is enclosed.

COMBINED TRANSMITTAL OF APPEAL BRIEF TO THE BOARD OF PATENT APPEALS AND INTERFERENCES & PETITION FOR EXTENSION OF TIME UNDER 37 C.F.R. 1.136(a) (Small Entity)	Docket No. WSS-10402/29

In Re Application Of: **McMillan et al**

Serial No.	Filing Date	Examiner	Group Art Unit
09/189,559	Nov. 11, 1998	Channavajjala	2155

Invention: **METHOD AND SYSTEM FOR MANAGING SOFTWARE CONFLICTS AND COMPUTER-READABLE STORAGE MEDIUM HAVING A PROGRAM FOR EXECUTING THE METHOD**

TO THE ASSISTANT COMMISSIONER FOR PATENTS:

This combined Transmittal of Appeal Brief to the Board of Patent Appeals and Interferences and petition for extension of time under 37 CFR 1.136(a) is respectfully submitted by the undersigned:

Signature

Dated: **Oct. 21, 2003**

John G. Posa
Reg. No. 37,424
Gifford, Krass, Groh
280 N. Old Woodward Ave., Suite 400
Birmingham, MI 48009
Tel. 734/913-9300
Fax 734/913-6007

Certificate of Transmission by Facsimile*

I certify that this document and authorization to charge deposit account is being facsimile transmitted to the United States Patent and Trademark Office (Fax. No.) on (Date)

Signature

Typed or Printed Name of Person Signing Certificate

***This certificate may only be used if paying by deposit account.**

Certificate of Mailing

I certify that this document and fee is being deposited on 10-21-03 with the U.S. Postal Service as first class mail under 37 C.F.R. 1.8 and is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231.

Sheryl L. Hammer

Signature of Person Mailing Correspondence

Sheryl L. Hammer

Typed or Printed Name of Person Mailing Correspondence

cc:

#28
11/5/03
A.W.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of: McMillan et al

Serial No.: 09/189,559 Group No.: 2177

Filed: Nov. 11, 1998 Examiner: Channavajjala

For: METHOD AND SYSTEM FOR MANAGING SOFTWARE CONFLICTS AND COMPUTER-READABLE STORAGE MEDIUM HAVING A PROGRAM FOR EXECUTING THE METHOD

APPELLANT'S BRIEF UNDER 37 CFR §1.192

Mail Stop Appeal Brief
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Dear Sir:

I. Real Party in Interest

The real party and interest in this case is Wise Solutions, Inc., a Michigan corporation, by assignment.

II. Related Appeals and Interferences

There are no appeals or interferences which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. Status of Claims

The present application was filed with 32 claims. Claims 33-65 were added in September, 2002. Accordingly, claims 1-65 are pending in this application. All are under appeal.

IV. Status of Amendments Filed Subsequent to Final Rejection

No after-final amendments have been filed.

GIFFORD, KRASS, GROH, SPRINKLE, ANDERSON & CITKOWSKI, P.C. 280 N. OLD WOODWARD AVENUE, STE. 400, BIRMINGHAM, MICHIGAN 48009-5394 (248) 647-6000

V. Concise Summary of the Invention

The invention resides in a method and system for managing software conflicts and computer-readable storage media having a program for executing the method wherein a database of interrelated tables is utilized (Specification, page 4, lines 23-26). Such conflicts may be identified before software applications are introduced to a computer system, thus improving user productivity by eliminating system outages and downtime (Specification, page 5, lines 1-5).

A preferred embodiment includes the steps of determining changes made to a computer system's files and other shared resources during installation of at least one application into the computer system to obtain change information (Specification, page 5, lines 6-10); processing the change information to determine which files and shared resources conflict with one another to obtain conflict information (Specification, page 5, lines 10-12); and storing the conflict information in a database of interrelated tables and resolving the software conflicts based on the stored conflict information (Specification, page 5, lines 12-14).

The applicable software conflicts may include file conflicts, registry conflicts, shortcut conflicts, ODBC driver conflicts, ODBC data source conflicts, service conflicts, device conflicts, component conflicts, autoexec.bat conflicts, config.sys conflicts, INI changes conflicts, and path conflicts (Specification, page 5, lines 21-24).

VI. Concise Statement of Issues Presented For Review

1. Are claims 1, 5-8, 14-18, 22-25, 31-43, 45-52 and 58-65 unpatentable under 35 U.S.C. §103(a) over U.S. Patent No. 5,586,304 to Stupek, Jr. et al. in view of U.S. Patent No. 6,018,747 to Burns et al.?

2. Are claims 2-4, 19-21, 44 and 55 unpatentable under 35 U.S.C. §103(a) over U.S. Patent No. 5,586,304 to Stupek, Jr. et al. in view of U.S. Patent No. 6,018,747 to Burns et al., further in view of U.S. Patent No. 5,634,114 to Shipley?

3. Are claims 9-13, 26-30 and 53-54 unpatentable under 35 U.S.C. §103(a) over U.S. Patent No. 5,586,304 to Stupek, Jr. et al. in view of U.S. Patent No. 6,018,747 to Burns et al., further in view of U.S. Patent No. 5,842,024 to Choye et al.?

GIFFORD, KRASS, GROH, SPRINKLE, ANDERSON & CITKOWSKI, P.C. 280 N. OLD WOODWARD AVENUE, STE. 400, BIRMINGHAM, MICHIGAN 48009-5394 (248) 647-6000

4. Are claims 56-57 unpatentable under 35 U.S.C. §103(a) over U.S. Patent No. 5,586,304 to Stupek, Jr. et al. in view of U.S. Patent No. 6,018,747 to Burns et al., further in view of U.S. Patent No. 6,192,375 to Gross?

VII. Grouping of Claims for Each Ground of Rejection Which Appellant Contends

Appellant believes the following groups of claims represent patentably distinct inventions which should be given independent consideration on appeal:

Group I: Claims 1, 5-8, 14-18, 22-25, 31-43, 45-52 and 58-65 wherein claims 4-8, 14-18, 22-25, 31-43, 45-52 and 58-65 stand or fall with claim 1;

Group II: Claims 2-4, 19-21, 44 and 55, wherein claims 3-4, 19-21, 44 and 55stand or fall with claim 2;

Group III: Claims 9-13, 26-30 and 53-54, wherein claims 10-13, 26-30 and 53-54 stand or fall with claim 9; and

Group IV: Claims 56-57 which stand or fall as one.

VIII. Argument

A. Group I - Claims 1, 5-8, 14-18, 22-25, 31-43, 45-52 and 58-65 wherein claims 4-8, 14-18, 22-25, 31-43, 45-52 and 58-65 stand or fall with claim 1.

The claims of this grouping stand rejected under 35 U.S.C. §103(a) over Stupek, Jr. et al., U.S. Patent No. 5,586,304, in view of Burns et al., U.S. Patent No. 6,018,747. For the reasons set for herein below, Appellant contends that the Examiner has not met the burden of establishing *prima facie* obviousness.

The '304 reference to Stupek, Jr. et al. is directed to the upgrading of a computer system resource from an existing version of the resource to a later version of the resource. The method includes the steps of digitally storing upgrade information which identifies the later version and describes features of the later version relevant to one or more earlier versions of the resource; digitally storing in the computer information identifying the existing version, by computer, automatically determine which of the earlier versions is the existing version; and based on the results of the comparing step (determining which of the

earlier versions is the existing version), automatically determining, or displaying to a user at least some of the upgrade information to aid the user in determining whether to perform an upgrade (see U.S. Patent No. 5,586,304, column 1, lines 56-67 to column 2, line 1).

The upgrade information according to Stupek, Jr. et al. may include information concerning reasons for the later version, and an indication of the type of change from a prior version to a later version (e.g. feature enhancement, performance enhancement, or bug fix) and/or an indication of the importance of the change from the prior version to the later version (see '304 Patent at column 2, lines 24-30). Apparently, this upgrade information is merely descriptive and is clearly not equivalent to the information provided to a user of Appellant's invention wherein *actual conflict information* is derived from comparisons made between files and shared resources of the new application with existing files and resources of the computer system.

U.S. Patent No. 6,018,747 to Burns et al., on the other hand, discloses a method of reconstructing a version of a file in places where data storage space and memory resources are limited. Two versions of the same dates or files are utilized and differencing methods are used to find the changes between the two files. As shown in Figure 6 of the Burns Patent, and as described at column 8, lines 42-47, conflicts occur when the delta file commands require the delta file to first write to a region of memory and then later read from that same region which results in the system comparing original data with altered data resulting in conflicts.

Appellant's claim 1 includes the steps of: receiving change information regarding actual changes made to files and other shared resources during installation of different applications into the computer system; processing the change information to determine conflict information pertaining to which files and shared resources conflict with one another; storing the conflict information in a database; and resolving any software conflicts based on the stored conflict information.

It is well-settled that "[o]bviousness may not be established using hindsight or in view of the teachings or suggestions of the invention." Para-Ordnance Mfg. Inc. v. SGS Importers Int'l Inc., 73 F.2d at 1087, 37 USPQ2d at 1239 (citing W.L. Gore & Assoc., Inc. v. Garlock Inc., 721 F.2d at 1551, 1553, 220 USPQ at 311, 312-313). In rejecting claims under 35 U.S.C. §103, the Examiner must provide a reason why one having ordinary skill in the pertinent art would have been led to combine references to arrive at Applicant's claimed invention. Moreover, there must be something *in the prior art* that suggests

GIFFORD, KRASS, GROH, SPRINKLE, ANDERSON & CITKOWSKI, P.C. 280 N. OLD WOODWARD AVENUE, STE. 400, BIRMINGHAM, MICHIGAN 48009-5394 (248) 647-6000

the proposed combination, other than the hindsight gained from knowledge that the inventor choose to combine these particular things in this particular way. Uniroyal Inc. v. Rudkin-Wiley Corp., 837 F.2d 1044, 1051, 5 USPQ2d 1434, 1438 (Fed. Cir. 1988). The Examiner is further required to make specific findings on a suggestion to combine prior-art references. In Re Dembeczak, 175 F.3d 994, 1000-01, 50 USPQ2d 1614, 1617-19 (Fed. Cir. 1999).

The Examiner has not met these burdens in this case. Not only is the prior art lacking in a teaching a motivation to combine the references, but even if the combination were made, the result would not read on this group of claims.

According to the '304 Patent to Stupek, Jr. et al., upgrade information is used to perform *only two types of comparisons:* (a) whether or not a particular upgrade package corresponds to a resource on the server, and (b) whether or not the version number of the upgrade package matches the version number of the corresponding network resource. This is falls short of the detailed comparisons set forth in claim 1, wherein files and other shared resource of a new application are compared to the files and shared resources of the existing applications of a computer system during installation to determine actual conflicts. The '304 reference simply does not teach or suggest a method of managing software conflicts in a computer system wherein the method includes a step of receiving change information regarding actual changes made to files and other shared resources during installation of different applications into the computer system and whereby the change information is used to determine conflict information pertaining to which files and shared resources conflict with one another after installation.

To the extent the '304 reference teaches the use of upgrade information to aid a user in determining whether or not to upgrade an existing system with a new application wherein the upgrade information does not include information relative to actual conflicts that exist between files or other shared resources between the new application and the existing applications in the computer system, Appellant's invention as claimed in this grouping of claims provides a more specific approach to determining the differences between a new application and existing applications in a computer system such that the user is provided a more informative view as to how the installation of a new application will affect the existing computer system.

Clearly the 304' reference is deficient in terms of forming the basis of an obviousness rejection. The addition of the Burns Patent does not help. As with Stupek, Jr., there is no suggestion in Burns of the

method of managing software conflicts as according to Appellant's invention wherein detailed comparisons are made between the files and shared resources of a new application and preexisting applications in a computer system to determine actual conflict information.

It has long been recognized that obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention unless there is some teaching, suggestion or incentive in the prior art which would have made such a combination appropriate. *Ashland Oil Inc. v. Delta Resins and Refractories Inc. et al.*, 227 USPQ 657, 667. Further, to establish prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. MPEP 8th Ed., 2143.03. Appellant submits that there is no teaching or suggestion of all the claim limitations of the present invention in the above cited references and therefore an obviousness rejection is improper. Accordingly, respectfully requests that this be withdrawn as a basis for rejection.

B. Group II - Claims 2-4, 19-21, 44 and 55, wherein claims 3-4, 19-21, 44 and 55 stand or fall with claim 2.

The claims of this grouping stand rejected under 35 U.S.C. §103 as being unpatentable over Stupek, Jr. et al., Burns et al., and further in view of Shipley, U.S. Patent No. 5,634,114. The goal of Shipley is to ensure that there will always be an expected data and control interface between a Dynamic Link Library (DLL) and an application program that relies on that DLL. The application program makes an initial call to the DLL that specifies the DLL version with which it prefers to operate. The DLL compares this preferred version number to the version numbers that it supports, which are in a supported DLL version table within the DLL. If the preferred version matches one of the table entries, then the DLL returns an "OK" flag to the application program. In this case, the application program goes on with its normal execution. However, if the preferred version is not supported, then the DLL returns to the application program a "not OK" flag and a list of versions that it does support from its table of supported versions. In response, the application program looks up the versions on this list in a table of compatible versions contained within the application program. If none are found, then the application performs an error trap. If one is found, then the application program calls the DLL to establish that version as the one which will be used. A header file is used to designate the DLL versions.

GIFFORD, KRASS, GROH, SPRINKLE, ANDERSON & CITKOWSKI, P.C. 280 N. OLD WOODWARD AVENUE, STE. 400, BIRMINGHAM, MICHIGAN 48009-5394 (248) 647-6000

Thus, while Shipley does address DLL versions, the addition of Shipley adds nothing in furtherance of the Examiner's argument that independent claim 1 is obvious over the Stupek, Jr. et al./Burns et al. combination. Shipley does not receive change information regarding actual changes made to files and other shared resources during installation of different applications into the computer system; nor does Shipley process the change information to determine conflict information pertaining to which files and shared resources conflict with one another, or store the conflict information in a database. Rather, according to Shipley, the application program makes an initial call to specify the DLL version with which it prefers to operate. The DLL compares this preferred version number to the version numbers that it supports by referencing a "supported DLL version table." If the preferred version is not supported, then the DLL returns to the application program a "preferred version not supported" flag and a list of versions that it does support. In response, the application program compares this list with the versions with which it is compatible; that is, it looks up each version on this list in its own "compatible version" table.

While the system of Shipley purportedly "prevents a later modified application program from calling an earlier version of the DLL," this is unrelated to the building of a potential conflict database during installation of different applications into the computer system. If an independent claim is not obvious under 35 U.S.C. §103, then any claim depending therefrom is non-obvious. *In re Fine*, 5 USPQ2d 1596 (Fed. Cir. 1988). In this case, the dependent claims depend from independent claims which are believed to be in condition for allowance.

C. Group III - Claims 9-13, 26-30 and 53-54, wherein claims 10-13, 26-30 and 53-54 stand or fall with claim 9.

The claims of this group stand rejected under 35 U.S.C. §103 as being unpatentable over Stupek, Jr. et al. in view of Burns et al. and further in view of Choye et al., U.S. Patent No. 5,842,024. Claim 9 adds to claim 1 the limitation of conflict information including Microsoft Windows Installer component conflict information. The disclosure of Choye et al. adds nothing in support of the Examiner's rejection of independent claim 1. Accordingly the claims of this group are allowable as well.

But in addition, the teachings of Choye et al. are non-analogous. Choye is directed to a method of reducing the time needed for installing a combination of software programs onto a hard disk drive.

This in done by creating modules for each software program and associating a file with each software program that records all changes made to files and directories by the installation process of that software program. When downloading combinations of these modules onto the hard disk drive, a software installation program is run which implements the changes to files and directories necessitated by each of the modules thereby resulting in a disk operating system that is properly configured for the operation of the combination of software programs. Although this involves Microsoft Windows, the process is unrelated to conflict negotiation and, in fact, assumes that conflicts regarding files and other shared resources will not arise.

D. Group IV -- Claims 56-57 which stand or fall as one.

Claims 56 and 57 are rejected under 35 U.S.C. §103(a) as being unpatentable of Stupek, Jr. et al. in view of Burns et al. and further in view of Gross, U.S. Patent No. 6,192,375. The claims of this group are specifically directed to conflict based on installer component GUIDs (global unique identifiers). Although the Gross reference mentions GUIDs, it is not in the context of a pre-determined conflict management environment. According to Gross, "[t]he storage medium 131 includes a third file 230 that has a dependency on a fourth file 240. The third file 230 is linked to the fourth file 240 via a registry file 250. A registry file 250 is a data base used by an operating system to store configuration information, file content, and file location information. The registry file 250 includes a path name file 251 that stores the file location information. The third file 230 includes a reference to a global unique identifier (GUID) 231 that identifies data required by the third file 230."

The disclosure of Gross adds nothing in support of the Examiner's rejection of independent claim 1. Accordingly the claims of this group are allowable as well.

Conclusion

In conclusion, for the arguments of record and the reasons set forth above, all pending claims of the subject application continue to be in condition for allowance and Appellant seeks the Board's concurrence at this time.

Respectfully submitted,

By: _____

Date: Oct. 21, 2003

John G. Posa
Reg. No. 34,424
Gifford, Krass, Groh, Sprinkle,
Anderson & Citkowski, P.C.
280 N. Old Woodward, Suite 400
Birmingham, MI 48009
(734) 913-9300

GIFFORD, KRASS, GROH, SPRINKLE, ANDERSON & CITKOWSKI, P.C. 280 N. OLD WOODWARD AVENUE, STE. 400, BIRMINGHAM, MICHIGAN 48009-5394 (248) 647-6000

APPENDIX A

CLAIMS ON APPEAL

1. A method of managing software conflicts in a computer system, comprising the steps of:

receiving change information regarding actual changes made to files and other shared resources during installation of different applications into the computer system;

processing the change information to determine conflict information pertaining to which files and shared resources conflict with one another;

storing the conflict information in a database; and

resolving any software conflicts based on the stored conflict information.

2. The method of claim 1, wherein the conflict information includes DLL file conflict information.

3. The method of claim 1, wherein the conflict information includes registry conflict information.

4. The method of claim 1, wherein the conflict information includes shortcut conflict information.

5. The method of claim 1, wherein the conflict information includes driver conflict information.

6. The method of claim 1, wherein the conflict information includes data source conflict information.

7. The method of claim 1, wherein the conflict information includes service conflict information.

8. The method of claim 1, wherein the conflict information includes device conflict information.

9. The method of claim 1, wherein the conflict information includes Microsoft Windows Installer component conflict information.

10. The method of claim 1, wherein the conflict information includes autoexec.bat conflict information.

11. The method of claim 1, wherein the conflict information includes config.sys conflict information.

12. The method of claim 1, wherein the conflict information includes INI changes conflict information.

13. The method of claim 1, wherein the conflict information includes path conflict information.

14. The method of claim 1, wherein the step of resolving any software conflicts includes the step of generating an installer based on the stored conflict information.

15. The method of claim 1, wherein at least one of the tables has a conflict field for storing a conflict level indication therein.

16. A computer-readable storage medium having stored therein a program which executes the steps of:

receiving change information regarding actual changes made to files and other shared resources during installation of different applications into the computer system;

processing the change information to determine conflict information pertaining to which files and shared resources conflict with one another;

storing the conflict information in a database; and

resolving any software conflicts based on the stored conflict information.

17. The storage medium of claim 16, wherein the program further executes the step of generating an installer for the application.

18. A system for managing software conflicts, comprising:

means for receiving change information regarding actual changes made to a computer system's files and other shared resources during installation of different applications into the computer system;

means for processing the change information to determine conflict information pertaining to which files and shared resources conflict with one another;

a database for storing the conflict information; and

means for resolving the software conflicts based on the stored conflict information.

19. The system of claim 18, wherein the conflict information includes DLL file conflict information.

20. The system of claim 18, wherein the conflict information includes registry conflict information.

21. The system of claim 18, wherein the conflict information includes shortcut conflict information.

22. The system of claim 18, wherein the conflict information includes driver conflict information.

GIFFORD, KRASS, GROH, SPRINKLE, ANDERSON & CITKOWSKI, P.C. 280 N. OLD WOODWARD AVENUE, STE. 400, BIRMINGHAM, MICHIGAN 48009-5394 (248) 647-6000

23. The system of claim 18, wherein the conflict information includes data source conflict information.

24. The system of claim 18, wherein the conflict information includes service conflict information.

25. The system of claim 18, wherein the conflict information includes device conflict information.

26. The system of claim 18, wherein the conflict information includes Microsoft Windows Installer component conflict information.

27. The system of claim 18, wherein the conflict information includes autoexec.bat conflict information.

28. The system of claim 18, wherein the conflict information includes config.sys conflict information.

29. The system of claim 18, wherein the conflict information includes INI changes conflict information.

30. The system of claim 18, wherein the conflict information includes path conflict information.

31. The system of claim 18, wherein the means for resolving includes means for generating an installer from the information stored in the database for the at least one application.

32. The system of claim 18, wherein at least one of the tables has a conflict field for storing a conflict level indicator therein.

33. A method of managing software conflicts in a computer system, comprising the steps of:

receiving change information regarding actual changes made to files and other shared resources during installation of at least one application into the computer system;

processing the change information to determine conflict information pertaining to which files and shared resources conflict with one another, the conflict information including one of a plurality of different conflict severity values;

storing the conflict information in a database; and

resolving any software conflicts based on the stored conflict information.

34. The method of claim 33, wherein the plurality of different conflict severity values comprises an informational value and an error value.

35. The method of claim 34, wherein the error value indicates a more severe conflict than the informational value.

36. The method of claim 34, wherein the plurality of different conflict severity values further comprises a warning value.

37. The method of claim 36, wherein the error value indicates a more severe conflict than the warning value, and wherein the warning value indicates a more severe conflict than the informational value.

38. The method of claim 36, wherein the database of tables includes a file conflict table having records only for files whose conflict value is either the warning value or the error value.

39. The method of claim 33 wherein, if two files have the same destination filename and destination directories, the conflict severity value is derived by:

determining if at least one of a version, an internal version, a date, a time and a size for the two

GIFFORD, KRASS, GROH, SPRINKLE, ANDERSON & CITKOWSKI, P.C. 280 N. OLD WOODWARD AVENUE, STE. 400, BIRMINGHAM, MICHIGAN 48009-5394 (248) 647-6000

files do not match.

40. The method of claim 33 wherein, if two files are 16-bit executables and have the same destination filename, the conflict severity value is derived by:

determining if at least one of a version, an internal version, a date, a time and a size for the two files do not match.

41. The method of claim 33 wherein, if two files have the same destination filename and destination directories, the conflict severity value is derived by:

determining if the two files have a version, an internal version, a date, a time and a size that match.

42. The method of claim 33, wherein, if two files are 16-bit executables and have the same destination filename, the conflict severity value is derived by:

determining if the two files have a version, an internal version, a date, a time and a size that match.

43. The method of claim 33 wherein, if two files are 16-bit executables and have the same destination filename, the conflict severity value is derived by:

determining if the destination directories for the two files do not match.

44. The method of claim 33 wherein, if two registry files have the same registry path and registry key names, the conflict severity value is derived by:

determining if at least one of a subset of characters of a text name of the registry key, a data type, and an operation for the two registry files do not match.

45. The method of claim 33 wherein, if two ODBC drivers have the same driver name and the same number of bits, the conflict severity value is derived by:

determining if an attribute exists for one of the two ODBC drivers and is non-existent for

another of the two ODBC drivers.

46. The method of claim 33 wherein, if two ODBC drivers have the same driver name and same number of bits, the conflict severity value is derived by:

determining if an attribute for one of the two ODBC drivers has a different value than the attribute for another of the two ODBC drivers.

47. The method of claim 33, including at least three conflict severity values.

48. The method of claim 33, wherein the conflict severity value is derived by determining a respective conflict value for each conflict.

49. The method of claim 33, wherein the step of storing the conflict information comprises storing the conflict value in the database.

50. The method of claim 33, wherein the conflict severity values include a warning value and an error value.

51. The method of claim 50, wherein the error value indicates a more severe conflict than the warning value.

52. The method of claim 33, wherein the plurality of different conflict severity values comprises an informational value, a warning value and an error value, wherein the error value indicates a more severe conflict than the warning value, and the warning value indicates a more severe conflict than the informational value, and

for each of two files having the same destination filename:

the conflict value is set to the warning value if destination directories for the two files match or the two files are 16-bit executables, and at least one of a version, an internal version, a date, a time and a size for the two files do not match;

the conflict value is set to the error value if the destination directories for the two files match or the two files are 16-bit executables, and the two files have a version, an internal version, a date, a time and a size which match; and

the conflict value is set to the informational value if the destination directories for the two files do not match and the two files are not 16-bit executables.

53. The method of claim 1, wherein two INI files conflict if they have the same path, section and entry.

54. The method of claim 1, wherein two configuration files conflict if they have the same command data and entry data.

55. The method of claim 1, wherein two shortcuts conflict if they have matching desktop-displayable human-readable text associated therewith, a matching location, and a matching icon group, but have either an unmatched path that leads to a program executed by the shortcut, unmatched parameters passed to the program being launched from the shortcut, an unmatched directory within which the launched program is to startup, an unmatched icon number for a particular program, or an unmatched path and filename of a file containing a desktop-displayable icon associated with the shortcut.

56. The method of claim 1, wherein resources are determined to conflict based on installer component GUIDs.

57. The method of claim 56, wherein identical resources are determined to conflict if they have different component GUIDs.

58. The method of claim 1, further including the step of determining the change information.

59. The method of claim 1, wherein the database is in the form of interrelated tables.

GIFFORD, KRASS, GROH, SPRINKLE, ANDERSON & CITKOWSKI, P.C. 280 N. OLD WOODWARD AVENUE, STE. 400, BIRMINGHAM, MICHIGAN 48009-5394 (248) 647-6000

60. The storage medium of claim 16, wherein the program is further operative to execute the step of determining the change information.

61. The storage medium of claim 16, wherein the database is in the form of interrelated tables.

62. The system of claim 18, further including means for determining the change information.

63. The system of claim 18, wherein the database is in the form of interrelated tables.

64. The method of claim 33, further including the step of determining the change information.

65. The method of claim 33, wherein the database is in the form of interrelated tables.